

## 2.13 DarBoot

In (fast) jedem Ozm-XMAX ROM ist [cecekpawons DarBoot](#) eingebaut. Es handelt sich dabei im Prinzip um einen eigenständigen Bootloader (DarBoot), für einen anderen "Bootloader" (Ozmosis).

DarBoot übernimmt die Aufgabe, verloren gegangene Darwin (macOS) Bootoptionen, wieder zur Bootauswahl hinzuzufügen. Dies ist besonders praktisch auf Mainboards, bei denen nach einem NVRam reset manche APFS Booteinträge verschwinden. Hier springt DarBoot ein und fügt die verlorenen Bootoptionen mit dem vom User definierten Namen wieder ins Bootmenü ein.

In jedem XMAX ROM mit integriertem DarBoot, ist ebenfalls eine DarBoot.plist enthalten. Diese Plist steuert den Treiber und lässt einiges an Konfiguration zu.

Die im ROM enthaltene Plist sieht so aus:

DarBoot.plist aus dem XMAX ROM

Um DarBoot nun auf die eigenen Bedürfnisse anzupassen, kann eine eigene DarBoot.plist in das erste Verzeichnis der EFI Partition gelegt werden. (Also nach `/Efi`, mit dem Namen `DarBoot.plist`)

Die vollständige DarBoot.plist sieht so aus:

Code

1. <dict>
2. <key>DarwinDiskTemplate</key>
3. <string></string>
4. <key>DarwinRecoveryDiskTemplate</key>
5. <string></string>
6. <key>DarwinInstallerDiskTemplate</key>
7. <string></string>
8. <key>DeleteInvalidPath</key>
9. <false/>
10. <key>BootMenu</key>
11. <false/>
12. <key>Timeout</key>
13. <integer>5</integer>
14. <key>DeleteEntries</key>
15. <array>
16. <string>Preboot</string>
17. </array>
18. <key>HideEntries</key>
19. <array>
20. <string>Recovery</string>
21. </array>
22. <key>Preferences</key>
23. <dict>
24. <key>Debug</key>
25. <false/>

26. <key>Off</key>
27. <false/>
28. <key>SaveLogToFile</key>
29. <true/>
30. <key>SaveLogToDeviceTree</key>
31. <false/>
32. <key>SaveLogToNvram</key>
33. <false/>
34. </dict>
35. </dict>

Alles anzeigen

- DarwinDiskTemplate legt den Namen von macOS-Installations Bootoptionen fest
- DarwinRecoveryDiskTemplate legt den Namen von macOS-Recovery Bootoptionen fest
- DarwinInstallerDiskTemplate legt den Namen von macOS-Installstick Bootoptionen fest

Der Eintrag für DarwinDiskTemplate könnte z.B. so aussehen: `$label $version`

Das Resultat davon wäre bei Mojave dann z.B. der Booteintrag-Name *Mojave 10.14*. Es wird hier also kein konkreter Name festgelegt (würde auch gehen), sondern der Aufbau des Booteintrags beschrieben. Möglich sind folgende Bestandteile:

- \$label = Name des Volumes (zB *Macintosh HD*, oder *Mojave*, oder *MeineFestplatte*)
  - \$path = Pfad zum Bootfile (zB `\2728FB71-33BF-4262-96BC-B6B4FF5C441C\System/Library\CoreServices`)
  - \$platform = Plattform auf welcher das OS läuft (zB Darwin)
  - \$version = OS Version (zB *10.14* für Mojave)
  - \$build = Build-Number des OS (zB 17G65)
  - \$major, \$minor und \$revision = OS Versions-Nummern (zB bei Mojave *10.14.2*: \$major = *10*, \$minor = *14*, \$revision = *2*)
- 
- DeleteInvalidPath kann aktiviert werden, wenn im Bootmenü manche Einträge doppelt erscheinen, die Option löscht ungültige Booteinträge
  - BootMenu aktiviert ein sehr simples DarBoot Bootmenü und Timeout legt die Anzeigzeit dieses Bootmenüs fest (nicht mit der Oz-GUI zu verwechseln)
  - Mit DeleteEntries lassen sich Booteinträge löschen. Heißt eine ungewünschte Bootoption im Bootmenü beispielsweise "Preboot", kann man hier einfach *Preboot* eintragen und der Eintrag wird gelöscht
  - Mit HideEntries lassen sich Booteinträge verstecken. Das ganze funktioniert wie DeleteEntries, die Booteinträge sind jedoch noch über das BIOS ausführbar und nicht komplett gelöscht
  - Debugaktiviert eine erweiterte Log-Ausgabe
  - Wird Off auf true gesetzt, ist DarBoot deaktiviert (also immer auf false/NO stehen lassen)
  - Alle weiteren SaveLog... Optionen aktivieren das Speichern des Logs z.B. ToFile, also auf die EFI Partition (SaveLogToFile sollte aktiviert werden)

Weitere Infos zu Bootargumenten für DarBoot und Optionen, gibts auf der offiziellen UEFTW Seite und dessen Wiki: <https://github.com/cecepawon/UEFTW>